An Efficient Multi-channel Distributed Coordination Protocol for Collaborative Engineering Activities

(Extended Abstract)

Saul Pomares Hernández, Khalil Drira, Jean Fanchon, Michel Diaz
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS
7, Av. du colonel Roche - 31077 Toulouse – France
{spomares, khalil, fanchon, diaz}@laas.fr

Key words: *Distributed engineering*, *GroupWare, Group communication.*

The advances in several computing fields, such as: networking (topology, bandwidth, etc), powerful data processing, and storage, have enabled in the last years, a great progress in collaborative technology. This collaborative technology is used as a support for advanced collaborative applications [1] such as distributed engineering. The distributed engineering applications consider *activities* that involve geographically distributed working *groups*, which interact into work *sessions* according to a preset or improvised planning. Such activities have been analyzed during the European Distributed System Engineering (DSE) project [2] where we made use of collaboration scenarios, which involved participants distributed in three sites, at Turin, Munich and Paris. During these sessions, participants needed to review design issues in collaboration with remote partners. One of the main problems that we found is to synchronously use, in a consistent manner, distributed engineering components and Conferencing GroupWare, such as the collaborative scenario depicted in Fig. 1a. This figure shows a scenario involving four participants using an Engineering CAD tool to share design data, a Conferencing GroupWare to discuss and comment design diagrams, and a Collaborative authoring tool to produce and review design documents. In this scenario, the participants that interact through the Conferencing Groupware component and Engineering CAD tool must have the same consistent view with respect to the shared design data in order to have a coherent discussion with the ensemble of participants. Otherwise, the comments about the design data would not make sense. This is a simple example of inter-component dependencies that must be satisfied in a consistent manner (Fig. 1b).

In this paper we propose a new fully distributed coordination protocol that avoids erroneous collaboration scenarios in distributed engineering components and applications. To achieve global coherence, our approach maintains two consistency levels: at a channel level and at an inter-channel level. In this way, we guarantee consistency for monolithic applications or components that use a unique communication channel, and consistency between components and applications that use multi-channel communication. This approach has been adopted as a result of the experience of the DSE project.

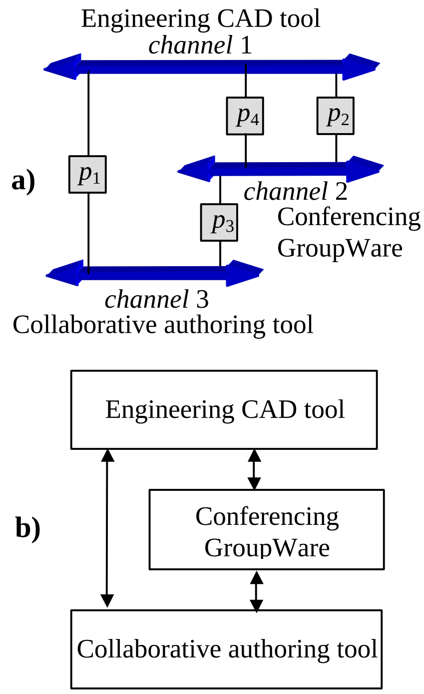

Fig. 1 a) Participant channel distribution,
b) Component dependencies diagram

Our approach supports an arbitrary number of participants and an arbitrary number of components. Both, participants and components can vary in time. In order to allow greater flexibility, the number of components used can be different between participants. Our coordination protocol allows a participant to interact with everybody in a non-halt manner. It is based on the causal principle and on multicast asynchronous channels.

The multi-channel coordination protocol

Our protocol manages application actions using an *event* representation. For example, an event can be: writing on a replicated design object, modifying a shared document, arrival of a new collaborator, etc. We manage two types of events: *intra-channel events* and *inter-channel events*. To ensure intra-*channel* consistency (for short *channel* consistency), all the channel events are diffused in causal relationship. To ensure inter-channel consistency (for short *global consistency*), our protocol detects and diffuses, in a causal manner, only the information about events that must traverse the channel frontier. At the end, our approach aims to maintain the independence of the channels as much as possible.

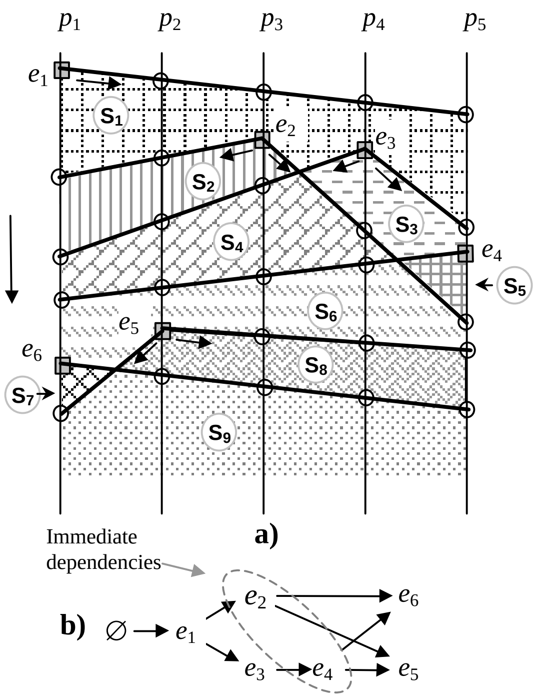

Fig. 2 One-channel a) Causal State Diagram, b) IDR graph

In order to allow participants to interact, we associate (without loss of generality) one multicast channel per component (See Fig. 1a). All the participants that share and use the same channel form a *collaboration group*.

To minimise the causal overhead, the coordination protocol uses the principle of *Immediate Dependency Relation* (IDR) (Definition 1), first, for the diffusion of intra-channel events, and second, for the diffusion of inter-channel events.

Definition 1. *Immediate dependency relation* $\downarrow$:

$$e \downarrow e' \Leftrightarrow [(e \rightarrow e') \wedge \forall e'' \in E, \neg (e \rightarrow e'' \rightarrow e')]$$

where E is the set of all possible events.

In order to understand the above principle regarding intra-channel events, consider the diffusion of event e_6 such that $(e_2 \| (e_3 \downarrow e_4)) \downarrow e_6$, see Fig. 2a. The question arises: which events have immediate dependency with e_6? Applying Definition 1, we found that the only events with IDR to e_6 are e_2 and e_4 (see Fig. 2b), and therefore, the only necessary causal information (CI) attached to the diffusion of e_6, in order to maintain channel consistency, corresponds to events e_2 and e_4. In a general way, we show in Fig. 2a that in a given time, a sub-set of participants can have the same sub-set of events with IDR. In this case, we say that the participants share a same causal state S_w. All participants that share a causal state have the same consistent partial view.

The multi-channel case is a little more complex. So as to better understand the diffusion of events between channels, we have constructed a three-dimensional diagram (Fig. 3), where each face of the diagram corresponds to one channel, and each vertex corresponds to one participant and its interactions with more than one channel. Briefly, we show that in our approach, the only information that must traverse the channel frontier belongs to the last local channel events with IDR. In the example shown in Fig. 3, the CI attached in the diffusion of event e_4 in channel 3 belongs to events e_2 and e_3. This information must be diffused in event e_5 to ensure global consistency.

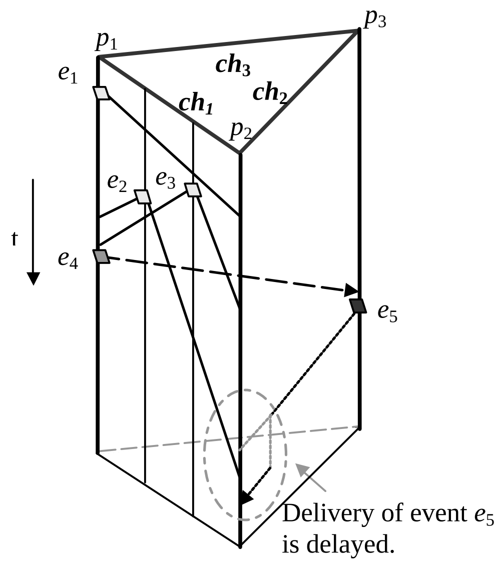

Fig. 3 Multi-channel 3D diagram

The full paper will present a complete description of our distributed coordination protocol, as well as its base principles and architecture. We will also present a complete bibliography and related work.

References

[1] H.-P. Dommel and J.J. Garcia-Luna-Aceves, Network Support for Group Coordination, 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000), Orlando, FL.

[2] K.Drira, A. Martelli and T.Villemur, Cooperative Environments for Distributed Systems Engineering, Lecture Notes in Computer Science 2236, Springer, 279 p., N°ISBN 3-540-43083-0, 2001.